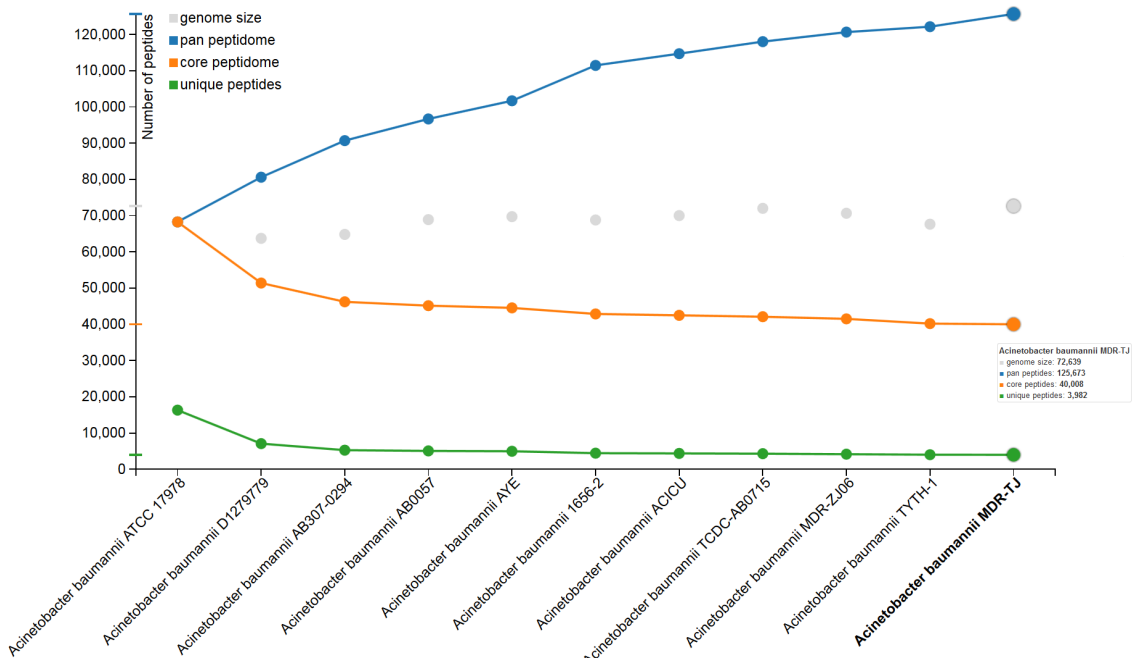
Filter peptides

Peptides are molecules which consist of a small chain of amino acids that are interconnected by means of a peptide bond. The Unique Peptide Finder application from the Unipept platform allows you to determine the set of **unique peptides** for a selected set of complete genomes. The application considers peptides to be unique if they occur in each of the selected genomes and in no other known genomes (except those belonging to the same species as the selected genomes).



The Unique Peptide Finder from the Unipept suite finds almost 4000 peptides that are unique to the bacterial species *Acinetobacter baumannii*. These peptides can be used as biomarkers to check whether or not the species appears in an environmental sample.

These unique peptides can then be used to determine whether an environmental sample contains traces of a given species, for example to assure that there is no horse meat in the meatballs from a Swedish furniture giant. Indeed, it is possible to tweak mass spectrometers for targeting certain peptides (*targeted proteomics*). To do so, additional selection criteria are usually imposed on the peptides to increase the possibility that they are picked up by the mass spectrometer. For example, with *selected reaction monitoring* (SRM) — a particular technique for *targeted proteomics* — only peptides having a length between 10 and 15 amino acids are used, and peptides containing cysteine (C), methionine (M), histidine (H) or tryptophan (W) are excluded. After all, during proteomics experiments cysteines are often reduced and alkalized to break sulfur bridges, and methionine, histidine and tryptophan have a tendency to become oxidized.

Assignment

In this assignment, peptides are represented as strings that only contain letters (in practice only 20 letters are used, corresponding to the 20 different amino acids). Your task is to implement a function filterPeptides that takes the location of two text files as string arguments. The first text file must contain a list of peptides, with each peptide on a separate line. This is the file format used by the Unique Peptide Finder to export lists of unique peptides. The function must write all peptides from this text file that meet certain criteria to a new text file (each peptide must be written on a separate line), whose location is passed to the function as its second argument. The criteria

that are applied to filter peptides can be set using four optional parameters of the function filterPeptides.

- minlen: an integer that indicates the minimal length of the peptide
- maxlen: an integer that indicates the maximal length of the peptide
- contains: a string whose letters represent amino acids that must all occur in the peptide
- lacks: a string whose letters represent amino acids that may not occur in the peptide

A filter is only applied if a value is passed to the corresponding parameter of the function. All filters that compare letters should work case insensitive. The peptides that meet all criteria should be written to the new text file without any modification.

As a helper function for the implementation of the function filterPeptides, you must first write a function filterPeptide. This function takes a peptide as an argument. In addition, the function has the same optional parameters as the function filterPeptides, with the same semantics. The function must return a Boolean value that indicates whether or not the given peptide meets all criteria.

Example

In the following interactive session we assume that the text file peptides.txt is located in the current directory.

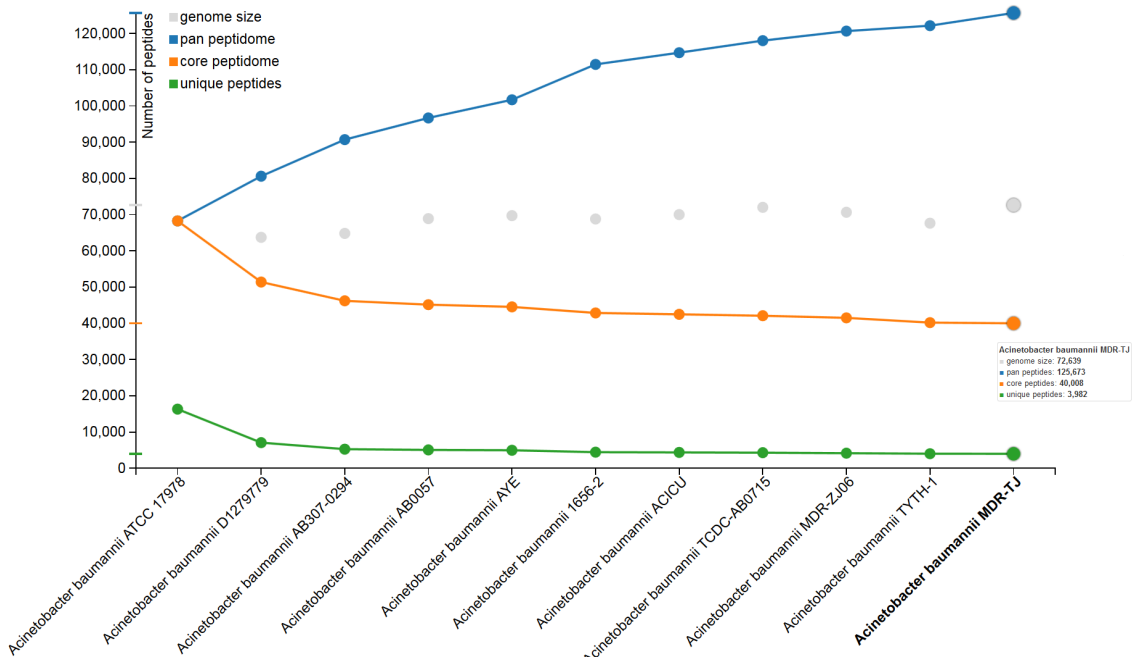
```
>>> filterPeptide('QEWLEMPWDNWPVYVLR', minlen=10, maxlen=20)
True
>>> filterPeptide('LICLSYGCHMMSYQWAHIVTDDCVDEGCGMYHMSHEILK', maxlen=20)
False
>>> filterPeptide('EQEETISFADLGPNGTFISK', contains='DEQ', lacks='CMHW')
True
>>> filterPeptide('QEWLEMPWDNWPVYVLR', contains='DEQ', minlen=10, maxlen=20, lacks='CMHW')
False

>>> filterPeptides('peptides.txt', 'filtered.txt', minlen=10, maxlen=20)
>>> print(open('filtered.txt', 'r').read().rstrip())
EQEETISFADLGPNGTFISK
QEWLEMPWDNWPVYVLR

>>> filterPeptides('peptides.txt', 'filtered.txt', maxlen=20)
>>> print(open('filtered.txt', 'r').read().rstrip())
ISIK
GLIR
EQEETISFADLGPNGTFISK
QEWLEMPWDNWPVYVLR

>>> filterPeptides('peptides.txt', 'filtered.txt', contains='DEQ', lacks='CMHW')
>>> print(open('filtered.txt', 'r').read().rstrip())
EQEETISFADLGPNGTFISK
SATIDLGIYTIADLAISGGTTDNVDGTGDAPGLGDIQEVPR

>>> filterPeptides('peptides.txt', 'filtered.txt', lacks='CMHW')
>>> print(open('filtered.txt', 'r').read().rstrip())
ISIK
GLIR
EQEETISFADLGPNGTFISK
SATIDLGIYTIADLAISGGTTDNVDGTGDAPGLGDIQEVPR

>>> filterPeptides('peptides.txt', 'filtered.txt', contains='DEQ', minlen=10, maxlen=20, lacks='CMHW')
```

```
>>> print(open('filtered.txt', 'r').read().rstrip())
EQEETISFADLGPNGTFISK
```

References

- **Lange V, Picotti P, Aebersold R (2008)**. Selected reaction monitoring for quantitative proteomics: a tutorial. *Molecular Systems Biology* **4**, 222.
- **Marx V (2013)**. Targeted proteomics. *Nature Methods* **10**, 19-22.
- **Mesuere B, Devreese B, Debyser G, Aerts M, Vandamme P, Dawyndt P (2012)**. Unipept: tryptic peptide-based biodiversity analysis of metaproteome samples. *Journal of Proteome Research* **11(12)**, 5773-5780.

Peptiden zijn moleculen opgebouwd uit een kleine keten van aminozuren die onderling verbonden zijn door middel van een peptidebinding. De Unique Peptide Finder toepassing van het Unipept platform laat je toe om een verzameling van **unieke peptiden** te bepalen voor een reeks geselecteerde genomen. De toepassing vat unieke peptiden op als peptiden die voorkomen in elk van de geselecteerde genomen, en in geen enkel ander genoom (behalve diegene die tot dezelfde soorten behoren als de geselecteerde genomen).



De Unique Peptide Finder uit de Unipept suite vindt bijna 4000 unieke peptiden voor de bacteriële soort *Acinetobacter baumannii*. Deze peptiden kunnen als biomerker gebruikt worden om na te gaan of deze soort in een staal voorkomt.

Deze unieke peptiden kunnen dan gebruikt worden om na te gaan of een omgevingsstaal sporen bevat van een bepaalde soort, bijvoorbeeld om na te gaan of er paardenvlees in de gehaktballen van een Zweedse meubelgigant zit. Massaspectrometers kunnen immers specifiek ingesteld worden om gericht bepaalde peptiden op te sporen (*targeted proteomics*). Hierbij worden bijkomende selectiecriteria opgelegd aan de peptiden, om daarmee de kans te vergroten dat ze door de massaspectrometer opgepikt zullen worden. Bij *selected reaction monitoring* (SRM) — een bepaalde techniek voor *targeted proteomics* — wordt bijvoorbeeld enkel gewerkt met peptiden die een lengte tussen 10 en 15 aminozuren hebben, en worden ook peptiden uitgesloten die cysteïne (C), methionine (M), histidine (H) of tryptofaan (W) bevatten. Cysteïnes worden in proteomics experimenten immers vaak gereduceerd en gealkalyseerd om zwavelbruggen te verbreken. Methionine, histidine en tryptofaan kunnen dan weer geoxideerd

worden.

Opgave

In deze opgave worden peptiden voorgesteld als strings die enkel bestaan uit letters (in de praktijk worden er slechts 20 letters gebruikt, die corresponderen met de 20 verschillende aminozuren). Je wordt gevraagd om een functie filterPeptiden te schrijven, waaraan twee locaties van tekstbestanden als stringargumenten moeten doorgegeven worden. Het eerste tekstbestand moet een lijst van peptiden bevatten. Elke peptide staat daarbij op een afzonderlijke regel. Dit is meteen ook het bestandsformaat waarmee de unieke peptiden uit de Unique Peptide Finder toepassing kunnen geëxporteerd worden. De functie moet alle peptiden uit het eerste bestand die voldoen aan bepaalde voorwaarden wegschrijven naar een nieuw tekstbestand (elke peptide moet op een afzonderlijke regel uitgeschreven worden), waarvan de locatie als tweede argument aan de functie wordt doorgegeven. De extra voorwaarden op basis waarvan de peptiden moeten gefilterd worden, kunnen ingesteld worden aan de hand van vier optionele parameters van de functie filterPeptiden.

- minlen: een natuurlijk getal dat de minimale lengte van een peptide aangeeft
- maxlen: een natuurlijk getal dat de maximale lengte van een peptide aangeeft
- bevat: een string waarvan de letters aminozuren voorstellen die allemaal in een peptide moeten voorkomen
- ontbreekt: een string waarvan de letters aminozuren voorstellen waarvan er geen enkel in het peptide mag voorkomen

Een filter wordt pas actief als er een waarde voor de corresponderende parameter aan de functie wordt doorgegeven. Alle filters die letters vergelijken, mogen geen onderscheid maken tussen hoofdletters en kleine letters. De peptiden die aan alle voorwaarden voldoen, moeten ongewijzigd naar het nieuwe bestand geschreven worden.

Als hulpfunctie bij de implementatie van de functie filterPeptiden schrijf je eerst een functie filterPeptide. Aan deze functie moet een peptide als argument doorgegeven worden. Voorts heeft de functie dezelfde optionele parameters als de functie filterPeptiden, met dezelfde betekenis. De functie moet een Booleaanse waarde teruggeven, die aangeeft of de gegeven peptide voldoet aan de gestelde voorwaarden.

Voorbeeld

Bij onderstaande voorbeeldsessie gaan we ervan uit dat het tekstbestand peptiden.txt zich in de huidige directory bevindt.

```
>>> filterPeptide('QEWLEMPWDNWPVYVLR', minlen=10, maxlen=20)
True
>>> filterPeptide('LICLSYGCHMMSYQWAHIVTDDCVDEGCGMYHMSHEILK', maxlen=20)
False
>>> filterPeptide('EQEETISFADLGPNGTFISK', bevat='DEQ', ontbreekt='CMHW')
True
>>> filterPeptide('QEWLEMPWDNWPVYVLR', bevat='DEQ', minlen=10, maxlen=20, ontbreekt='CMHW')
False

>>> filterPeptiden('peptiden.txt', 'gefilterd.txt', minlen=10, maxlen=20)
>>> print(open('gefilterd.txt', 'r').read().rstrip())
EQEETISFADLGPNGTFISK
```

```
QEWLEMPWDNWPVYVLR

>>> filterPeptiden('peptiden.txt', 'gefilterd.txt', maxlen=20)
>>> print(open('gefilterd.txt', 'r').read().rstrip())
ISIK
GLIR
EQEETISFADLGPNGTFISK
QEWLEMPWDNWPVYVLR

>>> filterPeptiden('peptiden.txt', 'gefilterd.txt', bevat='DEQ', ontbreekt='CMHW')
>>> print(open('gefilterd.txt', 'r').read().rstrip())
EQEETISFADLGPNGTFISK
SATIDLGIYTIADLAISGGTTDNVDGTGDAPGLGDIQEVPR

>>> filterPeptiden('peptiden.txt', 'gefilterd.txt', ontbreekt='CMHW')
>>> print(open('gefilterd.txt', 'r').read().rstrip())
ISIK
GLIR
EQEETISFADLGPNGTFISK
SATIDLGIYTIADLAISGGTTDNVDGTGDAPGLGDIQEVPR

>>> filterPeptiden('peptiden.txt', 'gefilterd.txt', bevat='DEQ', minlen=10, maxlen=20, ontbreekt='CMHW')
>>> print(open('gefilterd.txt', 'r').read().rstrip())
EQEETISFADLGPNGTFISK
```

Bronnen

- **Lange V, Picotti P, Aebersold R (2008)**. Selected reaction monitoring for quantitative proteomics: a tutorial. *Molecular Systems Biology* **4**, 222.
- **Marx V (2013)**. Targeted proteomics. *Nature Methods* **10**, 19-22.
- **Mesuere B, Devreese B, Debyser G, Aerts M, Vandamme P, Dawyndt P (2012)**. Unipept: tryptic peptide-based biodiversity analysis of metaproteome samples. *Journal of Proteome Research* **11(12)**, 5773-5780.